International Journal of Engineering and Geosciences
https://dergipark.org.tr/en/pub/ijeg
e-ISSN 2548-0960

# Deep learning-based vehicle detection from orthophoto and spatial accuracy analysis

**Muhammed Yahya Biyik [1], Muhammed Enes Atik [*1], Zaide Duran [1]**

[1]*Istanbul Technical University, Geomatics Engineering Department, Türkiye*

**Keywords**
UAV
Photogrammetry
Deep Learning
Orthophoto
Object Detection

Research Article
DOI: 10.26833/ijeg.1080624

Received: 28.02.2022
Accepted: 24.06.2022
Published: 19.10.2022

**Abstract**

Deep Learning algorithms are used by many different disciplines for various purposes, thanks to their ever-developing data processing skills. Convolutional neural network (CNN) are generally developed and used for this integration purpose. On the other hand, the widespread usage of Unmanned Aerial Vehicles (UAV) enables the collection of aerial photographs for Photogrammetric studies. In this study, these two fields were brought together and it was aimed to find the equivalents of the objects detected from the UAV images using deep learning in the global coordinate system and to evaluate their accuracy over these values. For these reasons, v3 and v4 versions of the YOLO algorithm, which prioritizes detecting the midpoint of the detected object, were trained in Google Colab's virtual machine environment using the prepared data set. The coordinate values read from the orthophoto and the coordinate values of the midpoints of the objects, which were derived according to the estimations made by the YOLO-v3 and YOLOV4-CSP models, were compared and their spatial accuracy was calculated. Accuracy of 16.8 cm was obtained with the YOLO-v3 and 15.5 cm with the YOLOv4-CSP. In addition, the mAP value was obtained as 80% for YOLOv3 and 87% for YOLOv4-CSP. F1-score is 80% for YOLOv3 and 85% for YOLOv4-CSP.

## 1. Introduction

Vehicle tracking and traffic situation analysis play an essential role in safe autonomous driving. Vehicle detection from unmanned aerial vehicle (UAV) imagery is a crucial task for many computers vision-based applications [1]. However, various factors associated with aerial photos, such as different vehicle sizes, orientations, types, density, limited datasets, and inference speed, make it a difficult process [2]. Many deep learning-based strategies have been developed in the literature in recent years to solve these issues [3].

When image processing techniques are combined with deep learning, successful techniques have begun to emerge in this regard [4]. Unmanned aerial vehicle (UAV) images are also widely used for object detection and tracking. UAVs have another concept increasing due to their customizability for different purposes and conditions. Considering these two solutions together is an inevitable unity, but it is possible to see their examples in many fields such as cultural heritage [5-7], building extraction [8, 9], mapping [10], geology [11], mining [12], shoreline extraction [13]. Images obtained with high sensitivity in very low flight using UAVs can be produced at a lower cost than images obtained from conventional aerial photogrammetry [10].

For computer vision and image processing, artificial intelligence (AI) offers new methodologies and benefits. Deep learning (DL) algorithms are becoming increasingly popular since they are resilient and require fewer human operations [14]. Deep learning has an essential role in many typical applications and it is gradually developing [15]. Deep Learning is essentially one of the sub-branches of artificial intelligence technology. Although Artificial Intelligence has been on the agenda since the 1950s, Deep Learning has been more effective since the 2010s [16]. The main reasons for this are the developing hardware features and the new algorithms and methods developed depending on the increasing amount of data. In simple terms, deep learning is the process of mathematically modeling the learning process of human beings.

* **Corresponding Author**

(biyik16@itu.edu.tr) ORCID ID 0000-0001-9848-9673
*(atikm@itu.edu.tr) ORCID ID 0000-0003-2273-7751
(duranza@itu.edu.tr) ORCID ID 0000-0002-1608-0119

**Cite this article**

Bıyık, M. Y., Atik, M. E., & Duran, Z. (2023). Deep learning-based vehicle detection from orthophoto and spatial accuracy analysis. International Journal of Engineering and Geosciences, 8(2), 138-145

Convolutional Neural Network (CNN) has been developed based on imitating the working principles of the neural networks and learning mechanism in the human brain [16]. Neural networks are capable of solving complicated problems with accuracy [17]. Object detection is a challenging topic in photogrammetry and computer vision. One of the primary issues is the need for human interaction in object detection [18]. CNN models are one of the important tools to reduce human intervention. CNN processes images used as input in layers for object detection. As a result of various filtering processes, the image becomes a flat matrix. As a result of the mathematical equations that are solved based on the parameters determined for the model using the resulting matrix, a weight and a bias value are determined for each node. Objects in the new images evaluated as a result of these weights are detected.

CNN-based object approaches are examined under two title: two-stage and single-stage detection methods [19]. Regions with convolutional neural networks (R-CNN), Fast R-CNN and Faster RCNN can be counted as two-stage detectors. R-CNN divides images into region proposals and applies CNN for each region respectively. CNN decides the appropriate region and object size. However, two-stage approaches stand out with their slowness. The speed problem is solved with single-stage approaches. You Only Look Once (YOLO) is a single-stage object detection approach. YOLO thinks that object detection is a regression problem and obtain the position of the object, category and corresponding confidence score. It increases the detection speed and detects the object in the real-time target [20].

The aim of this study is to consider the DL algorithms developed for detection from a photogrammetric perspective and to compare the midpoints of the detected objects using the produced position accuracies. Vehicle detection was performed over orthophoto using YOLOv3 and YOLOv4-CSP algorithms. The midpoint coordinates of the detected vehicles are obtained in the EPSG:5254 TUREF / TM30 coordinate system. Accuracy analysis was performed by comparing the detected coordinates and reference coordinates.

## 2. Literature Review

There are different neural network algorithms for object detection. For example, deep learning algorithms were used in these studies [19, 21, 22, 23].

In the study by Cepni et al. (2020) [19], vehicles were selected as the object to be detected and their data were analyzed according to the YOLOv3- YOLOv3-spp and YOLOv3-tiny algorithms, with the models they trained on Google Colab using the COCO data set and UAV images. Average IoU is obtained as 84,88% with YOLOv3-spp. Zhao et al. [21] presented a compact solution for vehicle detection, tracking and ground coordinate using a microcomputer integrated UAV system. They preferred YOLOv3 as a deep learning algorithm in the model they established in real-time. However, the model used was chosen as a pre-trained model due to the physical conditions of the UAV platform. They stated that they preferred a passive system to calculate the coordinates of the target object, so the positioning process was carried out according to the data obtained from the GPS and IMU sensors on the UAV. In the study, mAP value of 90.61% was reached with YOLOv3. Božić-Štulić et al. [24] aim to perform automatic object detection and positioning from aerial photographs using convolutional neural networks (CNN) without the need for Ground Control Points (GCP), in a three-stage proposed method as a new approach. The vehicle detection accuracy is 82.5% by using R-CNN. Liu et al. [25] have presented a solution called UAV-YOLO to solve the difficulties experienced in detecting small objects from UAV-based images in their study based on deep learning algorithms. They stated the study's aims as creating an image data set obtained from the UAV platform to improve the human detection performance and improve the neural network structure of the YOLO algorithm. In the study, the YOLOv3 algorithm was chosen and an improvement was made for the purpose of the study by using the Darknet software framework. In this way, model training and backbone structure optimization were provided. As a result of the study, a 90.86% mAP value was obtained in object detection made on UAV images. Zhang et al. [26] explains the stages of the approach they developed based on the problems experienced by object detection algorithms and integrated with the TrackleNet Tracker (TNT) method in their photograph and video series. In this approach, the RetinaNet method was preferred for object detection, as it was more convenient in the subsequent monitoring phase. In the multi-object detection phase, the TNT method is included to eliminate incomplete or unreliable detections that may arise. Subsequently, Visual Odometry and Ground Plane Estimation were made using the multi-image stereo (MWS) method. Finally, 3D object positioning was done from 2D image coordinates, thanks to the camera parameters and the data obtained in the previous processes. The detection performance of the study is 97.8% Map for cars. Radovic, et al. [27] determined their motivation as testing CNN object detection algorithms that can be used for autonomous UAV applications in civil engineering. They preferred the YOLO algorithm in training the artificial network. Images obtained using Google Maps platform were used to test the trained model. They obtained 97.5% accuracy using satellite images. In the study conducted by Atik et al. [15], it was aimed to detect different objects over DOTA data with YOLO-v2 and YOLO-v3 algorithms. The vehicles are detected with 59% F1-score using YOLOv3.

## 3. Material and Method

### 3.1. Dataset

In the study, two separate data sets were used for training and testing. The dataset used for the training consists of aerial images obtained with UAV on the Prince Sultan University campus. This dataset, published as open-source on GitHub, was prepared for vehicle detection from UAV-based images [28]. The platform height was specified as 55 m and 80 m while the images were collected (Fig. 1). The training set contains 218 images and 3,365 instances of labeled cars.

**Figure 1.** A sample image from training dataset

For the performance tests of the algorithms, aerial images obtained from the UAV flight over Istanbul Technic University Ayazaga Campus were used (Fig. 2). Forward overlap 80% and side overlaps of 70% and flown at a height of 80 m above were selected as flight parameters. Ground Sampling Distance (GSD) is around 3 cm/pixel. Vehicle detection was performed on the scaled and coordinated orthophotos created with these images. For accuracy analysis, vehicles were manually labeled and ground truth was produced.

**Figure 2.** A sample image from test dataset

### 3.2. You Only Look Once (YOLO)

You Only Look Once (YOLO) [20] is among the most well-known deep learning algorithms, and it stands out with its speed thanks to its single-stage detection architecture. In addition, if necessary, optimizations are made, it can work as CPU-based and is open source. Also, some pixels will have object conflict. Thanks to the "Anchor Box" approach in the YOLO algorithm, this problem can be solved.

There are some points to consider when using the YOLO algorithm. Because it processes images in one step, large images can exceed hardware capacities, and the detection accuracy of small objects in large images may be low. For this reason, the data to be used should be well examined. If necessary, large-size images should be divided into parts. Label files created as a result of the labeling process should be prepared in the format required by the YOLO algorithm. This format is shown in Table 1. The x and y coordinates, representing the centroid of the boxes drawn during labeling, take a value between 0 and 1. Width and height are values for the dimensions of the rectangle drawn for the label or bounding box (Fig. 3).

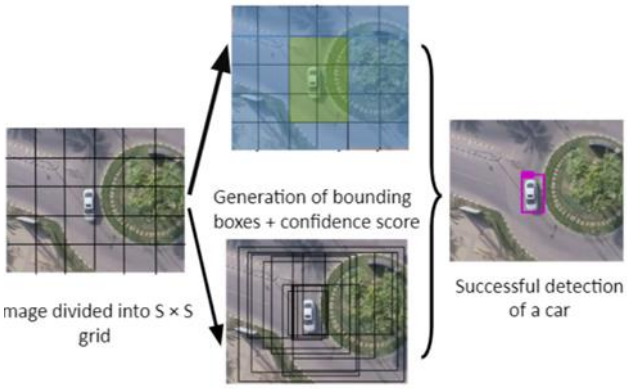


**Figure 3.** YOLO Algorithm work schema [28]

YOLO-v3 has residual skip connections and an up-sampling layer. Also, it makes detections at three different scales at layer 61, 94 and 106 [29]. It has darknet53 backbone [30]. Moreover, Rectified Linear Unit (ReLU) function was preferred as an activation function in YOLO-v3. In this study, YOLOv4-CSP (Cross-Spatial-Partial) is used. It has similar architect, but it uses the CSPdarknet53 backbone. YOLOv4-CSP [31] has cut-mix and mosaic data augmentation layers as default and without increasing training or detection time. YOLOv4-CSP uses a mish activation function and it has a Spatial Pyramid Pooling (SPP) module. The main difference between YOLO-v3 and YOLOv4-CSP algorithms is the backbone structure they use. Darknet53 is used for YOLO-v3, while CSPDarknet53 is used for YOLOv4-CSP. Accordingly, YOLO-v3 consists of a total of 106 layers, while YOLOv4-CSP contains a total of 161 layers.

The mathematical expression of the confidence score for the predictions made by the YOLO algorithm is specified in equation 2.1, as expressed in the article explaining the YOLO-v3 algorithm [30].

$$\Pr(Class_i|Object) * \Pr(Object) * IoU_{pred}^{truth} = \Pr(Class_i) * IoU_{pred}^{truth} \quad (1)$$

According to the equation, the confidence score of a prediction box is the product of the confidence score of the object that can be detected in it and the IOU value obtained for the box. It takes a value between 0 and 1. A high IOU value is an indicator of how accurately the dimensions of the detected object and the region it is located are detected.

### 3.3. Evaluation Metrics

As evaluation metrics, F1-score and mean average precision (mAP) are utilized. Precision is used to calculate the percentage of points that are classified as positive. The recall of a collection of positives is the percentage of true positives. The mean Average Precision, or mAP score, is the mean precision over all classes and/or overall IoU thresholds, depending on the numerous detection challenges that occur.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1\ score = 2\frac{Precision\ .\ Recall}{Precision + Recall} \quad (4)$$

$$AP = \sum_{k=0}^{k=n-1} [\text{Recalls(k)} - \text{Recalls(k} + 1)] * \text{Precisions(k)} \quad (5)$$

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \quad (6)$$

True positive (TP) is the number of points that are prediction is positive and the actual label is positive. False positive (FP) is the number of points that are prediction is positive and the actual label is negative. False negative (FN) is the number of points that are prediction is negative and the actual label is positive [32].

## 4. Experiment

The algorithms compared within the scope of the study were YOLO-v3 and YOLOv4-CSP algorithms. Both algorithms are prepared in accordance with the Darknet software framework. In order to compare the models, all parameters were kept constant during the training of the models. During the training of both models, parameter values published as open source by the publishers of Darknet Software Architecture were used. The workflow is presented in Fig. 4.

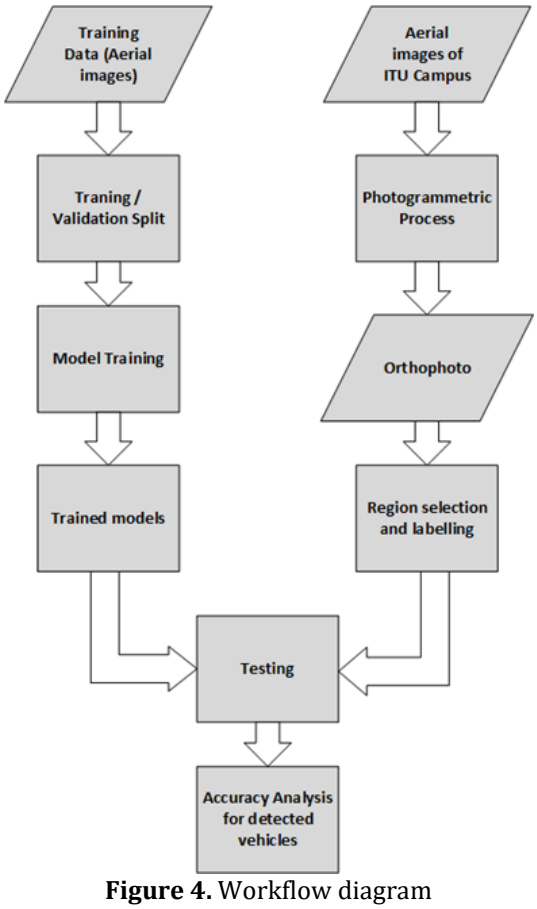


**Figure 4.** Workflow diagram

### 4.1. Producing Orthophoto

Photogrammetry is a scientific technique to calculate an object's three-dimensional coordinates by measuring the correspondence points in overlapping images and to provide reliable sensor calibrate for 3D object modelling from digital images [33]. As computer vision algorithms and photogrammetric technologies are combined, procedures that automate the image-based 3D modeling process become more common [34]. Agisoft Photoscan was used to model the point cloud. The software is based on the Structure-from-Motion (SfM) algorithm, which works on the basis of photogrammetry principles, in which the 3D coordinates of an object are determined by measuring the matching points between two overlapping photos.

Removing the errors due to topography in photographs and making them vertical is called orthorectification. The final product with coordinate information is called orthophoto. Coordinates of certain points are marked on the Orthophoto by using the GCP whose coordinates are known. As a result of balancing and interpolation processes, the coordinate values of any location on the orthophoto can be obtained. Although there are open source and paid software solutions for these operations, while producing the orthophoto, 372 images of 4000x3000 pixels were used (Fig. 5). The orthophoto has 3 cm/pixel spatial resolution. One of the factors affecting the accuracy of the orthophoto is the distribution of ground control points on the area. In the study, attention was paid to the homogeneous distribution of ground control points. In addition, care was taken to produce the generated point cloud at high density (83 pts/m$^2$). Thus, the digital elevation model (DEM) produced has high accuracy as a result of interpolation. DEM quality is one of the main factors affecting orthophoto accuracy.

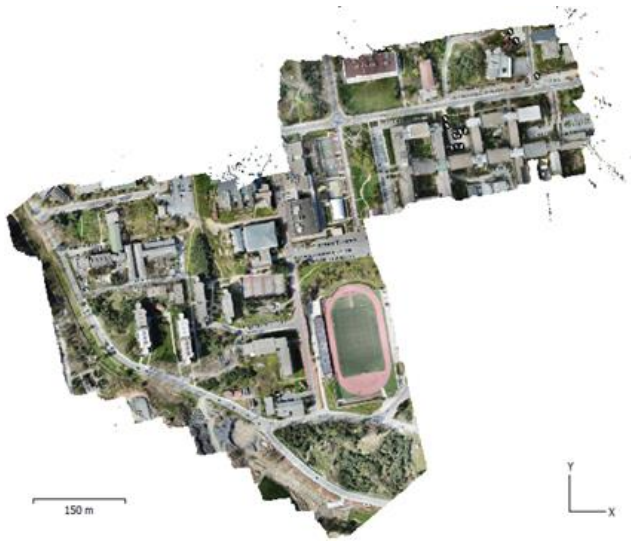


**Figure 5.** Orthophoto of Istanbul Technical University Ayazaga Campus.

The produced Orthophoto contains 10 ground control points (GCP). 3D coordinates of GCPs were measured by terrestrial measurements (using Global Navigation Satellite System (GNSS)). In addition, in order to calculate the accuracy of the model, 5 control points were

selected from the images in conditions that can be distinguished and intersect. The technical report of the model was prepared using Agisoft Professional software. The accuracy values obtained based on this report are shown in Table 1.

**Table 1.** Accuracy assessment result of check points.

| Point | X Error (cm) | Y Error (cm) | Z Error (cm) | Total (cm) |
|---|---|---|---|---|
| 3 | 0,2 | 0,5 | 0,1 | 0,5 |
| 6 | -3,7 | -1,0 | 5,6 | 6,8 |
| 7 | 2,4 | 0,0 | -5,7 | 6,2 |
| 8 | 2,0 | -0,2 | -5,7 | 6,1 |
| 9 | -0,4 | -0,4 | 1,7 | 1,8 |
| ∑ | 2,2 | 0,5 | 4,4 | 5,0 |

### 4.2. Training

Within the scope of the study, YOLO-v3 and YOLOv4-CSP models were trained. Model training was done in cloud environment using Google Colab. The training period varies depending on the hardware power. The training dataset divided into 90% training and 10% validation. Training parameters are given in Table 2.

**Table 2.** Training parameters.

| Parameter | YOLOv3 | YOLOv4-CSP |
|---|---|---|
| Batch Size | 16 | 16 |
| Subdivision | 4 | 4 |
| Width x height | 608x608 | 608x608 |
| Max. batch | 6000 | 6000 |
| Learning rate | 0.001 | 0.001 |
| Momentum | 0.9 | 0.9 |

During the training phase, the parameter files of the models were accepted as standard and no changes were made. The data used during model training are data shared as open source. The images are separated as training and validation data, the same distribution is used for each model. Although 2000 iterations are recommended for each number of classes, since there is only one class in our study, the models are trained with 6000 iterations, which is the minimum number of iterations recommended by the developers. The weights with the highest sensitivity value among the obtained weights were selected for the test. Trained models were tested on images selected over orthophoto.

**Table 3.** Analysis of position accuracy of vehicles detected on Figure 7, test area 2 using Model YOLO-v3

| | Orthophotos Coordinates (m) | | Detected Coordinates (m) | | Variation (m) | | | |
|---|---|---|---|---|---|---|---|---|
| Object | RIGHT | UP | RIGHT | UP | RIGHT | UP | Vx^2 | Vy^2 |
| 1 | 418093,865 | 4552668,461 | - | - | - | - | - | - |
| 2 | 418096,459 | 4552657,237 | 418096,443 | 4552657,497 | 0,016 | -0,260 | 0,000 | 0,068 |
| 3 | 418104,607 | 4552621,107 | - | - | - | - | - | - |
| 4 | 418109,700 | 4552640,200 | - | - | - | - | - | - |
| 5 | 418110,083 | 4552622,712 | 418110,131 | 4552622,884 | -0,048 | -0,172 | 0,002 | 0,029 |
| 6 | 418112,624 | 4552623,083 | - | - | - | - | - | - |
| 7 | 418115,305 | 4552641,617 | - | - | - | - | - | - |
| 8 | 418117,522 | 4552623,495 | - | - | - | | - | - |
| 9 | 418117,788 | 4552647,484 | 418118,028 | 4552647,399 | -0,240 | 0,085 | 0,058 | 0,007 |
| 10 | 418119,946 | 4552623,651 | - | - | - | - | - | - |
| 11 | 418126,056 | 4552648,435 | 418125,993 | 4552648,197 | 0,063 | 0,238 | 0,004 | 0,057 |
| | | | | | | | RMSE= | 0,168 m |

**Table 4.** Analysis of position accuracy of vehicles detected on Figure 7, test area 2 using Model YOLOv4-CSP

| | Orthophotos Coordinates (m) | | Detected Coordinates (m) | | Variation (m) | | | |
|---|---|---|---|---|---|---|---|---|
| Object | RIGHT | UP | RIGHT | UP | RIGHT | UP | Vx^2 | Vy^2 |
| 1 | 418093,865 | 4552668,461 | 418093,829 | 4552668,681 | 0,036 | -0,220 | 0,001 | 0,048 |
| 2 | 418096,459 | 4552657,237 | 418096,388 | 4552657,290 | 0,071 | -0,054 | 0,005 | 0,003 |
| 3 | 418104,607 | 4552621,107 | 418104,518 | 4552621,481 | 0,088 | -0,374 | 0,008 | 0,140 |
| 4 | 418109,700 | 4552640,200 | 418109,650 | 4552640,245 | 0,050 | -0,045 | 0,003 | 0,002 |
| 5 | 418110,083 | 4552622,712 | 418109,980 | 4552622,856 | 0,103 | -0,144 | 0,011 | 0,021 |
| 6 | 418112,624 | 4552623,083 | 418112,662 | 4552623,159 | -0,039 | -0,077 | 0,002 | 0,006 |
| 7 | 418115,305 | 4552641,617 | 418115,276 | 4552641,704 | 0,028 | -0,087 | 0,001 | 0,008 |
| 8 | 418117,522 | 4552623,495 | 418117,519 | 4552623,833 | 0,003 | -0,338 | 0,000 | 0,114 |
| 9 | 418117,788 | 4552647,484 | 418117,780 | 4552647,550 | 0,007 | -0,066 | 0,000 | 0,004 |
| 10 | 418119,946 | 4552623,651 | 418119,899 | 4552623,902 | 0,047 | -0,251 | 0,002 | 0,063 |
| 11 | 418126,056 | 4552648,435 | 418125,800 | 4552648,582 | 0,256 | -0,147 | 0,066 | 0,022 |
| | | | | | | | RMSE= | 0,155 m |

**Table 5.** Comparison of models using common detected vehicles on test area 2

| | Orthophotos Coordinates (m) | | Detected Coordinates (m) | | Variation (m) | | | |
|---|---|---|---|---|---|---|---|---|
| Object | RIGHT | UP | RIGHT | UP | RIGHT | UP | Vx^2 | Vy^2 |
| 2 | 418096,459 | 4552657,237 | 418096,388 | 4552657,290 | 0,071 | -0,054 | 0,005 | 0,003 |
| 5 | 418110,083 | 4552622,712 | 418109,980 | 4552622,856 | 0,103 | -0,144 | 0,011 | 0,021 |
| 9 | 418117,788 | 4552647,484 | 418117,780 | 4552647,550 | 0,007 | -0,066 | 0,000 | 0,004 |
| 11 | 418126,056 | 4552648,435 | 418125,800 | 4552648,582 | 0,256 | -0,147 | 0,066 | 0,022 |
| | | | | | | | RMSE= | 0,128 m |

## 5. Results

In the study, accuracy analysis was analyzed in two stages, the accuracy of the algorithm applied and the spatial accuracy of the detected vehicles.

Two regions were selected over orthophoto for evaluation. The detection performances of the algorithms were compared. In addition to the confidence values of the vehicles found by the algorithms, the number of vehicles they could detect was also evaluated. 0.80 mAP was achieved with Yolo-v3, while 0.87 mAP was achieved with YOLOv4-CSP. The YOLOv4-CSP has an F1 score of 0.85, while the YOLO-V3 has an F1 score of 0.80 (Fig. 6 and Fig. 7). The accuracy of the trained models is presented in Table 6. The confusion matrix is presented in Table 7 and Table 8.

**Table 6.** Statistical outputs of models

| Model | mAP | mIOU | F1 Score |
|---|---|---|---|
| YOLO-v3 | 0.80 | 0.55 | 0.80 |
| YOLOv4-CSP | 0.87 | 0.60 | 0.85 |

**Table 7.** The detection performance of the algorithms for sample images.

| Figure | Model | Detection Ratio | Confidence Score Mean |
|---|---|---|---|
| Test Area 1 | YOLO-v3 | 0,64 | 79.0 |
| Test Area 1 | YOLOv4-CSP | 1.0 | 99,18 |
| Test Area 2 | YOLO-v3 | 0,78 | 91.0 |
| Test Area 2 | YOLOv4-CSP | 1.0 | 98,78 |

**Table 8.** Confusion matrix of YOLOv4-CSP and YOLOv3

| | YOLOv4-CSP | | YOLOv3 | |
|---|---|---|---|---|
| | Predicted Positive | Predicted Negative | Predicted Positive | Predicted Negative |
| Actual Positive | TP=26 | FN=2 | TP=26 | FN=2 |
| Actual Negative | FP=1 | TN=0 | FP=1 | TN=0 |

**Figure 6.** Evaluation result on test area 1 (YOLO-v3 on the left; YOLOv4-CSP on the right)

**Figure 7.** Evaluation result on test area 2 (YOLO-v3 on the left; YOLOv4-CSP on the right).

The operator marked the midpoints of the detected vehicles to obtain reference values (Fig. 8). Then, the coordinates of the midpoints of the bounding boxes created by the algorithms are calculated. Thus, a spatial accuracy analysis was performed. Spatial accuracy analysis was applied separately for both all vehicles and common vehicles detected by both algorithms. Since all vehicles detected by YOLO-v3 were also detected by YOLOv4-CSP, only the accuracy of YOLOv4-CSP in common vehicles was analyzed. The purpose of preparing Table 4 is to show the position accuracy of the detected vehicles only for both models. Spatial accuracy for all tools is 16.8 cm for YOLOv3 and 15.6 cm for YOLOv4-CSP (Table 3 and Table 4). In common detected vehicles, the accuracy of YOLOv4 is 12.8 cm.


**Figure 8.** Obtaining the midpoints of the vehicles via Agisoft Metashape software

## 6. Discussion

As a result of the comparison between YOLO-v3 and YOLOv4-CSP, it was seen that the detection accuracy of the YOLOv4-CSP algorithm was higher in terms of both detection rate and spatial accuracy. YOLOv4-CSP can detect almost all vehicles in images. YOLO-v3 is unable to detect most vehicles close to image boundaries. The confidence value of the vehicles found is also higher in YOLOv4-CSP. The CSPDarknet architecture and the many layers make YOLOv4-CSP superior to YOLO-v3.

According to the accuracy analysis, the YOLOv4-CSP model detected the midpoints with higher accuracy than the YOLO-v3 model. Considering the common vehicles, vehicle positions can be detected with high accuracy with the YOLOv4-CSP. Bounding boxes are found by YOLOv4-CSP with higher accuracy. As seen in Table 6, the mIoU value is higher than that of YOLOv4-CSP. High mIoU in this case means high location accuracy. Appropriate data set and parameter selection in deep learning networks is important not only for detecting vehicles but also for obtaining accurate spatial information.

The reason for choosing YOLOv4-CSP among YOLOv4-CSP versions is to provide a healthier comparison with YOLO-v3. Thus, the effect of CSP optimization used in the backbone will be better understood. In addition, YOLOv4-CSP also requires less hardware resources and training time during training as it contains fewer layers than other YOLOv4-CSP versions. Finally, new YOLO versions are releasing with different approaches. According to their developers' claim some of them shows better mAP performance than the YOLOv4-CSP.

Consequently, However, as a point to be noted, the coordinate values reached do not express an absolute accuracy, it should be noted that the coordinate values taken from the orthophoto and accepted as correct have at least as much error as the orthophoto. However, since the same orthophoto was used during the study, this error amount was ignored assuming that it was equally distributed to each determination.

Area-based matching methods are also used for vehicle detection [35]. However, in area-based approaches, changes in the resolution or point of view of the image adversely affect the detection performance. It is more appropriate to use deep learning approaches instead of domain-based approaches, which are useful in cases where the search and target image are very similar. Especially in complex city structures, deep learning-based approaches are more suitable for vehicle detection. The YOLOv4-CSP algorithm correctly detected almost all vehicles in the images (Table 7).

In addition, training and test data were obtained from different sources in the study. This problem is called domain-shift in the literature. Models trained on aerial photographs obtained from a different region with a different UAV were tested on a completely different orthophoto with training data. It has been concluded that YOLOv4-CSP is less affected by the domain-shift problem and can be used in such cases.

## 7. Conclusion

In this study, we detect the vehicles from the orthophoto using different deep learning algorithms and analyze the spatial accuracy of the midpoints of the vehicles detected. It is possible that this work, which currently requires manual interventions and is carried out with non-instantaneous data, will become a fully autonomous and instantaneous solution in the future. In order to achieve this, it is a preferable solution to derive the positions of the detected objects depending on the UAV location, with the increase in the capabilities of the GNSS and IMU sensors on the UAVs.

In addition, the servicing of GNSS coordinates instead of limiting the results to image coordinates, especially in object detection studies with air platforms, will speed up and facilitate the production of location information, which many disciplines need.

Finally, new YOLO versions are releasing with different approaches. According to their developers' claim some of them shows better mAP performance than the YOLOv4-CSP. So, this study can be expanse with YOLO-v5 and YOLO-X.

### Author contributions

**Muhammed Yahya Biyik:** Data curation, Software, Field study, Writing-Original draft preparation, Visualization **Muhammed Enes Atik:** Methodology, Writing-Original draft preparation, Software, Validation, Conceptualization **Zaide Duran:** Conceptualization, Investigation, Writing-Reviewing and Editing, Supervision.

### Conflicts of interest

The authors declare no conflicts of interest.

### References

1. Luo, X., Tian, X., Zhang, H., Hou, W., Leng, G., Xu, W., ... & Zhang, J. (2020). Fast automatic vehicle detection in uav images using convolutional neural networks. *Remote Sensing*, 12(12), 1994.
2. Lu, J., Ma, C., Li, L., Xing, X., Zhang, Y., Wang, Z., & Xu, J. (2018). A vehicle detection method for aerial image based on YOLO. *Journal of Computer and Communications*, 6(11), 98-107.
3. Sang, J., Wu, Z., Guo, P., Hu, H., Xiang, H., Zhang, Q., & Cai, B. (2018). An improved YOLOv2 for vehicle detection. Sensors, 18(12), 4272.
4. Jazayeri, A., Cai, H., Zheng, J. Y., & Tuceryan, M. (2011). Vehicle detection and tracking in car video based on motion model. *IEEE Transactions on Intelligent Transportation Systems*, *12*(2), 583-595.
5. Senkal, E., Kaplan, G., & Avdan, U. (2021). Accuracy assessment of digital surface models from unmanned aerial vehicles' imagery on archaeological sites. *International Journal of Engineering and Geosciences*, *6*(2), 81-89.
6. Şasi A., & Yakar, M. (2018). Photogrammetric modelling of hasbey dar'ülhuffaz (masjid) using an unmanned aerial vehicle. *International Journal of Engineering and Geosciences*, *3*(1), 6-11.
7. Ulvi, A., Yakar, M., Yigit, A. Y., & Kaya, Y. (2020). İha ve Yersel Fotogrametrik Teknikler Kullanarak Aksaray

Kızıl Kilisenin 3b Modelinin ve Nokta Bulutunun Elde Edilmesi. *Geomatik*, *5*(1), 19-26.

8. Gönültaş, F., Atik, M. E. & Duran, Z. (2020). Extraction of Roof Planes from Different Point Clouds Using RANSAC Algorithm. *International Journal of Environment and Geoinformatics*, 7 (2), 165-171. https://doi.org/10.30897/ijegeo.715510
9. Atik, M. E., Donmez, S. O., Duran, Z., & İpbuker, C. (2018). Comparison Of Automatic Feature Extraction Methods for Building Roof Planes by Using Airborne Lidar Data and High-Resolution Satellite Image. *In Proceedings, 7th International Conference on Cartography and GIS (*pp. 857-863), 18-23 June 2018, Sozopol, Bulgaria.
10. Ulvi, A., Yakar, M., Yiğit, A. Y. & Kaya, Y. (2020). İha ve Yersel Fotogrametrik Teknikler Kullanarak Aksaray Kızıl Kilisenin 3b Modelinin ve Nokta Bulutunun Elde Edilmesi. Geomatik, 5 (1), 19-26. https://doi.org/10.29128/geomatik.560179
11. Agca. M., Gültekin, N., & Kaya, E. (2020). İnsansız hava aracından elde edilen veriler ile kaya düşme potansiyelinin değerlendirilmesi: Adam Kayalar örneği, Mersin. *Geomatik*, *5*(2), 134-145.
12. Alptekin, A. & Yakar, M. (2020). Determination of pond volume with using an unmanned aerial vehicle. Mersin Photogrammetry Journal, 2 (2), 59-63.
13. Aykut, N. O. (2019). İnsansız Hava Araçlarının Kıyı Çizgisinin Belirlenmesinde Kullanılabilirliğinin Araştırılması. *Geomatik*, *4*(2), 141-146.
14. Atik, S. O., & Ipbuker, C. (2021). Integrating Convolutional Neural Network and Multiresolution Segmentation for Land Cover and Land Use Mapping Using Satellite Imagery. *Applied Sciences*, *11*(12), 5551.
15. Atik, M. E., Duran, Z., & Özgünlük, R. (2022). Comparison of YOLO Versions for Object Detection from Aerial Images. *International Journal of Environment and Geoinformatics*, *9*(2), 87-93.
16. Bengio, Y., Goodfellow, I., & Courville, A. (2017). *Deep learning* (Vol. 1). Cambridge, MA, USA: MIT press.
17. Tasdemir, S., & Ozkan, I. A. (2019). ANN approach for estimation of cow weight depending on photogrammetric body dimensions. *International Journal of Engineering and Geosciences*, *4*(1), 36-44.
18. Sariturk, B., Bayram, B., Duran, Z., & Seker, D. Z. (2020). Feature extraction from satellite images using segnet and fully convolutional networks (FCN). *International Journal of Engineering and Geosciences*, *5*(3), 138-143.
19. Cepni, S., Atik, M. E., & Duran, Z. (2020). Vehicle detection using different deep learning algorithms from image sequence. *Baltic Journal of Modern Computing*, *8*(2), 347-358.
20. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).
21. Zhao, X., Pu, F., Wang, Z., Chen, H., & Xu, Z. (2019). Detection, tracking, and geolocation of moving vehicle from uav using monocular camera. *IEEE Access*, *7*, 101160-101170.
22. Atik, S. O., & Ipbuker, C. (2020). Instance Segmentation of Crowd Detection in The Camera Images. In *Proceeding of 41st Asian Conference on Remote Sensing 2020 (ACRS2020)*, Deqing City, Virtual, 9-11 November 2020.
23. Cazzato, D., Cimarelli, C., Sanchez-Lopez, J. L., Voos, H., & Leo, M. (2020). A survey of computer vision methods for 2d object detection from unmanned aerial vehicles. *Journal of Imaging*, *6*(8), 78.
24. Božić-Štulić, D., Kružić, S., Gotovac, S., & Papić, V. (2018). Complete model for automatic object detection and localisation on aerial images using convolutional neural networks. *Journal of Communications Software and Systems*, *14*(1), 82-90.
25. Liu, M., Wang, X., Zhou, A., Fu, X., Ma, Y., & Piao, C. (2020). Uav-yolo: Small object detection on unmanned aerial vehicle perspective. *Sensors*, *20*(8), 2238.
26. Zhang, H., Wang, G., Lei, Z., & Hwang, J. N. (2019, October). Eye in the sky: Drone-based object tracking and 3d localization. In *Proceedings of the 27th ACM International Conference on Multimedia* (pp. 899-907).
27. Radovic, M., Adarkwa, O., & Wang, Q. (2017). Object recognition in aerial images using convolutional neural networks. *Journal of Imaging*, *3*(2), 21.
28. Benjdira, B., Khursheed, T., Koubaa, A., Ammar, A., & Ouni, K. (2019, February). Car detection using unmanned aerial vehicles: Comparison between faster r-cnn and yolov3. In *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)* (pp. 1-6). IEEE.
29. Kim, S., & Kim, H. (2021). Zero-centered fixed-point quantization with iterative retraining for deep convolutional neural network-based object detectors. *IEEE Access*, *9*, 20828-20839.
30. Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
31. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.
32. Atik, M. E., Duran, Z., & Seker, D. Z. (2021). Machine learning-based supervised classification of point clouds using multiscale geometric features. ISPRS *International Journal of Geo-Information*, 10(3), 187.
33. Duran, Z., & Aydar, U. (2012). Digital modeling of world's first known length reference unit: The Nippur cubit rod. Journal of cultural heritage, 13(3), 352-356.
34. Duran, Z., & Atik, M. E. (2021). Accuracy comparison of interior orientation parameters from different photogrammetric software and direct linear transformation method. *International Journal of Engineering and Geosciences*, 6(2), 74-80.
35. Zhu, H., & Yu, F. (2016). A cross-correlation technique for vehicle detections in wireless magnetic sensor network. *IEEE Sensors Journal*, 16(11), 4484-4494.